Scale-space Feature Extraction on Digital Surfaces

Jérémy Levallois, David Coeurjolly, Jacques-Olivier Lachaud

Scale-space Feature Extraction on Digital Surfaces

Jérémy Levallois[a,b,*], David Coeurjolly[a], Jacques-Olivier Lachaud[b,c]

[a] *Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France*
[b] *Université de Savoie, CNRS, LAMA, UMR 5127, F-73776, France*
[c] *Université Grenoble-Alpes, CNRS, LJK, UMR 5224, F-38041, France*

Abstract

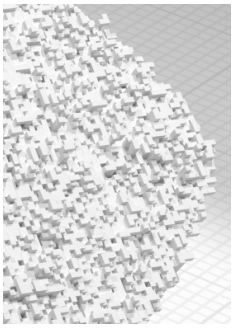
A classical problem in many computer graphics applications consists in extracting significant zones or points on an object surface, like loci of tangent discontinuity (*edges*), maxima or minima of curvatures, inflection points, etc. These places have specific local geometrical properties and often called generically *features*. An important problem is related to the scale, or range of scales, for which a feature is relevant. We propose a new robust method to detect features on digital data (surface of objects in $\mathbb{Z}^3$), which exploits asymptotic properties of recent digital curvature estimators. In [1, 2], authors have proposed curvature estimators (mean, principal and Gaussian) on 2D and 3D digitized shapes and have demonstrated their multigrid convergence (for C^3-smooth surfaces). Since such approaches integrate local information within a ball around points of interest, the radius is a crucial parameter. In this article, we consider the radius as a scale-space parameter. By analyzing the behavior of such curvature estimators as the ball radius tends to zero, we propose a tool to efficiently characterize and extract several relevant features (edges, smooth and flat parts) on digital surfaces.

Keywords: feature extraction, digital geometry, scale-space, curvature estimation, multigrid convergence, integral invariants.

1. Introduction

When performing geometry processing on shapes, a classical problem in many computer graphic applications consists in delineating places with specific local geometrical information — or *features* — on the shape surface. Even if no clear definition of feature on surface stands out, prior works usually characterize a feature as a local discontinuity distinguishable from its neighborhood. As an example, differential quantities have been widely considered in this context as preliminary information from which features can be extracted. However, an important problem is related to the scale (or range of scales) for which a feature is relevant. This question leads to scale-space analysis of shapes. Note that this concept has been widely investigated in the image processing community [3].

In this article, we propose a new robust feature extraction technique which incorporates scale-space geometrical information and which is dedicated to digital surfaces (boundary of objects in $\mathbb{Z}^2$ or $\mathbb{Z}^3$). We consider raw digital data (shapes discretized on a regular grid) as input for two main reasons: First, many acquisition devices (*e.g.* 3D MRI images or X-ray tomography) provide such data and we do not want to introduce approximations or interpolations by switching to a polyhedral representation. Second, working on digital data allows us to consider a mathematical framework — the multigrid convergence of operators — dedicated to this digital model. These digital surfaces are specific in the sense that boundaries of volumetric objects usually lead to a large number of surface elements. Furthermore, due to the digitization effect, digital surfaces can be considered as approximations of continuous manifolds with a very specific and isothetic noise model: samples are evenly spaced but never lie on the surface, normals are not informative. This kind of data could be problematic if it is not carefully handled when defining differential estimators for instance. Finally, this case study is motivated by accurate shape analysis of 3D volumetric porous material (microstructures of snow samples, see Fig. 7). In this context, we want to characterize geometrical discontinuities (edges) from smooth areas and zero curvature (flat) regions in a robust way.



Related works First of all, shape discontinuities can be formalized as *ridges and valleys* with differential geometry. In this case, such discontinuities are deduced from differential quantities of order 3 by looking at variations of principal curvature directions in a neighborhood [4, 5]. The final step consists in thresholding significant angular deviation of principal directions. Such techniques provide a formal approach to discontinuities extraction but are scale dependent and rely on a robust estimation of order 3 differentials. When dealing with noisy data or digital data, such approaches are not relevant and

☆This work has been mainly funded by DigitalSnow ANR-11-BS02-009 research grants.

*Corresponding author.
Email addresses: jeremy.levallois@liris.cnrs.fr (Jérémy Levallois), david.coeurjolly@liris.cnrs.fr (David Coeurjolly), jacques-olivier.lachaud@univ-savoie.fr (Jacques-Olivier Lachaud)

cannot be considered.

For meshes or point clouds, many approaches are based on integral quantities computed on local patches. For instance, Pauly *et al.* [6] and Clarenz *et al.* [7] have used Principal Component Analysis on data points located in a given neighborhood of the point of interest. A feature score is defined as a function of the eigenvalues of this covariance matrix. Then, either the feature score is simply thresholded, or the behavior of the score as a function of the neighborhood size is analyzed. Mérigot *et al.* [8] extended this approach to consider convolved covariance matrices of Voronoi cells (*Voronoi Covariance Measure* or VCM). Thresholding a ratio of VCM eigenvalues leads to a robust extraction of edges on point clouds or meshes. Such approaches produce interesting results at a fixed scale or for smooth objects. However, scale-space analysis is not fully integrated in these frameworks. Furthermore, even if ratios of covariance matrix eigenvalues are related to principal curvatures, the geometrical interpretation of scores is not straightforward. In the experimental section, we provide a more details on this approach.

In a similar way, Park *et al.* [9] have proposed a Tensor-Voting strategy on local surface patches. They used the scale-space behavior of the tensor vote when the neighborhood size increases, in order to extract edges on point clouds. As shown in the experiments, this technique is very sensitive and does not provide sufficiently robust results on digital surfaces. Mellado *et al.* [10] have introduced a fast least square spherical fitting approach to a point cloud to create a multi-scale feature score. Again, the scale-space parameter is the neighborhood size considered in the fitting. Even if this feature score is qualitatively relevant, it is not directly related to some geometrical information. Furthermore, when used on digital data, such technique fails to provide a precise localization of features.

Finally, features can be extracted following a spectral analysis of the shape from eigenvalues of the surface Laplacian matrix [11, 12, 13]. In this context, features are characterized by spectral quantities which are locally stable and distinguishable from its neighborhood. Such techniques are very promising but drawbacks exist for digital surfaces. First, since our surfaces have a large number of elements, computing the eigenvalues of the Laplacian matrix could be very computationally expensive. Another bottleneck relies on the fact that for digital surfaces, the isothetic nature of the Euclidean embedding (digitization on axis aligned grid) makes the metric not well embedded in the discrete Laplacian operator. Indeed, if we consider the DEC formulation or simply the *cotan* approach to define a discrete Laplacian operator on the digital surface embeddings, the staircase effect of the digitization makes the metric not well described by the geometrical embedding of the surface. For example, a consequence is that heat diffusion obtained by this operator produces anisotropic artifacts (ellipsoidal isocontours on a digital plane with normal vector $(1, 1, 0)^T$ for instance). On digital surfaces, a discrete Laplacian operator with correct intrinsic metric information has to be defined.

Contributions We propose a robust scale-space feature selector that classifies digital surface elements into three categories: *edge*, *smooth* or *flat*. This feature selector is built upon digital curvature estimators and relies on their theoretical multigrid convergence properties. Since these estimators are parametrized by the size of their ball of integration, *i.e.* a kind of scale, the feature selector analyses curvature estimated as function of scales. Since we know the theoretical behavior of models *edge*, *smooth* and *flat*, the feature selector chooses the model that best fits its input data. We compare our approach on a large class of shapes with the other above-mentioned approaches to feature detection, and we evaluate their robustness to noise. Finally, we apply this feature selector to the analysis of microstructures of 3D snow samples.

2. Preliminaries

In Geometry Processing, *integral invariants* have been widely investigated to construct estimators of differential quantities on smooth surface [14, 15]. The main idea is to move a ball B_R of radius R on points x of the boundary ∂X of shape X. Then, integrals are computed on the intersection between this ball and the shape, *i.e.* on $B_R(x) \cap X$ (see Fig. 1-a for notations). More formally, by Taylor expansion of the area and volume around the point x, 2D curvature estimator $\tilde{\kappa}_R(x)$ and 3D mean curvature estimator $\tilde{H}_R(x)$ can be defined respectively as [14]:

$$\tilde{\kappa}_R(X,x) \stackrel{def}{=} \frac{3\pi}{2R} - \frac{3A_R(x)}{R^3}, \quad \tilde{H}_R(X,x) \stackrel{def}{=} \frac{8}{3R} - \frac{4V_R(x)}{\pi R^4}, \quad (1)$$

where $X \subset \mathbb{R}^2$ (resp. $\mathbb{R}^3$) is a sufficiently smooth shape. Here $A_R(x)$ is the area and $V_R(x)$ the volume of $B_R(x) \cap X$ (*i.e.* we integrate the unit constant function on $B_R(x) \cap X$). $\tilde{\kappa}_R(X,x)$ and $\tilde{H}_R(X,x)$ values converge to expected ones (respectively curvature κ and mean curvature H) as R tends to zero [14], since:

$$\tilde{\kappa}_R(X,x) = \kappa(X,x) + O(R), \quad \tilde{H}_R(X,x) = H(X,x) + O(R). \quad (2)$$

Similarly, principal curvatures can be estimated by computing the two greatest eigenvalues λ_1 and λ_2 of the covariance matrix of $B_R(x) \cap X$ [14]:

$$\tilde{\kappa}^1(X,x) \stackrel{def}{=} \frac{6(\lambda_2 - 3\lambda_1)}{\pi R^6} + \frac{8}{5R} + O(R), \quad (3)$$

$$\tilde{\kappa}^2(X,x) \stackrel{def}{=} \frac{6(\lambda_1 - 3\lambda_2)}{\pi R^6} + \frac{8}{5R} + O(R). \quad (4)$$

Using similar integration principles, several estimators of various differential quantities can be defined. Please refer to [15, 16] for an overview.

2.1. Integral based Digital Curvature Estimators

In our context, we consider digital shapes (any subset of $\mathbb{Z}^d$) and boundaries of digital shapes. We denote by $\mathtt{D}_h(X)$ the Gauss digitization of X in a $d-$dimensional grid with grid step h, *i.e.* $\mathtt{D}_h(X) = X \cap (h\mathbb{Z})^d$. For such digitized set Z, $Bd(Z)$ denotes its topological boundary, seen as a cellular Cartesian complex (See Fig. 1-b). It is thus composed of 0-cells and 1-cells (resp. *pointels* and *linels*), and, for $d = 3$, with 2-cells (*surfels*), embedded in the digital grid.

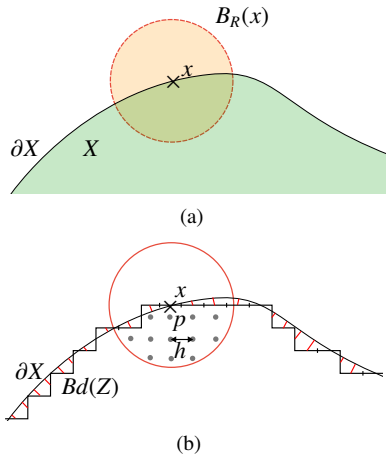


Figure 1: Integral invariant computation (*a*) and notations (*b*) in dimension 2 [1].

Before going further, we define the 2D digital curvature estimator $\hat{\kappa}_R$, the 3D digital mean curvature estimator $\hat{H}_R$ and the 3D digital principal curvature estimators κ^1_R and κ^2_R on $Z \subset \mathbb{Z}^2$ or $Z \subset \mathbb{Z}^3$:

Definition 1. *Given $Z \subset \mathbb{Z}^2$ (or $\mathbb{Z}^3$ for 3D estimators) and h a gridstep,* digital curvature estimators *are defined for any pointel $p \in Bd(Z)$ as:*

$$\hat{\kappa}_R(Z,p) \stackrel{def}{=} \frac{1}{h}\left(\frac{3\pi}{2R_d} - \frac{3A_{R_d}(p)}{{R_d}^3}\right), \tag{5}$$

$$\hat{H}_R(Z,p) \stackrel{def}{=} \frac{1}{h}\left(\frac{8}{3R_d} - \frac{4V_{R_d}(p)}{\pi {R_d}^4}\right), \tag{6}$$

$$\hat{\kappa}^1(Z,p) \stackrel{def}{=} \frac{1}{h}\left(\frac{6(\hat{\lambda}_2 - 3\hat{\lambda}_1)}{\pi {R_d}^6} + \frac{8}{5R_d}\right), \tag{7}$$

$$\hat{\kappa}^2(Z,p) \stackrel{def}{=} \frac{1}{h}\left(\frac{6(\hat{\lambda}_1 - 3\hat{\lambda}_2)}{\pi {R_d}^6} + \frac{8}{5R_d}\right), \tag{8}$$

where $R_d = \frac{R}{h}$ is the digitized radius of the ball, and $A_{R_d}(p)$ and $V_{R_d}(p)$ are the number of digital points in the intersection between Z and the ball. Similarly, $\hat{\lambda}_1$ and $\hat{\lambda}_2$ are the two greatest eigenvalues of the covariance matrix of the digital points in the intersection between Z and the ball.

Multigrid convergence results have been established in [1] for convex shapes with at least C^3-boundary and bounded curvature. The idea of multigrid convergence is that when we define a geometric quantity estimator on $\mathbb{D}_h(X)$, the estimate should converge (theoretically and experimentally) to the expected geometric quantity on X when the digitization step h gets finer and finer (*i.e.* tends to zero). Proofs of convergence for these digital curvature estimators rely on the fact that both R and h tends to zero, but at a well chosen speed. Authors of [1] have proved the convergence of these estimators when setting the ball radius dependent on the grid step h: $R = kh^{\frac{1}{3}}$, where k is a constant related to the maximal curvature of the shape.

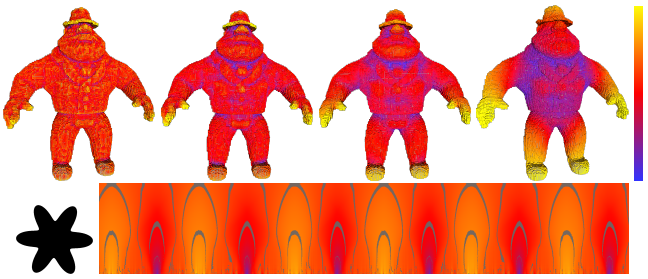


Figure 2: *Top*: Mean curvature using Eq. 6 for different ball radii: 4, 7, 14 and 30. *Bottom*: 2D flower and curvature values on its boundary using Eq. 5 (x-axis) for a range of decreasing ball radii (y-axis, from top to bottom). Curvature values are mapped from blue (lower) to yellow (higher), discrete gray lines illustrate curvature isovalues.

More formally, we get:

$$\hat{\kappa}_R(\mathbb{D}_h(X),p) = \kappa(X,x) + O(h^{\frac{1}{3}}), \tag{9}$$

$$\hat{H}_R(\mathbb{D}_h(X),p) = H(X,x) + O(h^{\frac{1}{3}}), \tag{10}$$

$$\hat{\kappa}^1_R(\mathbb{D}_h(X),p) = \kappa^1(X,x) + O(h^{\frac{1}{3}}), \tag{11}$$

$$\hat{\kappa}^2_R(\mathbb{D}_h(X),p) = \kappa^2(X,x) + O(h^{\frac{1}{3}}), \tag{12}$$

when the ball radius is $R = kh^{\frac{1}{3}}$.

When estimating a differential quantity, we usually have to specify either a window, a kernel size or a ball radius in which the computations are performed ([15, 10] for example). In other words, to process a shape at a given resolution, we need to choose a suitable radius R to capture relevant geometrical features. In figure 2-top we display the mean curvature estimations (cf. Eq. 6) on a digital object with different ball sizes. As expected, small radii are more sensitive to noise or digitization effects while large radii smoothen results. Therefore, the radius R specifies at which scale the quantity becomes relevant according to a given noise level. It follows that we should study a shape at several scales to obtain more representative geometric information.

To sum up, recent digital curvature estimators are well suited to digital data and contain interesting mathematical properties when processing digitizations of shapes with at least C^3-boundary and bounded curvature. In next section, we combine curvature estimations in a range of scales and these multigrid convergence properties to construct a feature selector for digital objects, which decides whether a point of its boundary falls into either *edge* (non$-C^1$ parts), *smooth* (smooth C^3 parts) or *flat* (zero curvature parts) categories.

3. Feature detection

We propose to study the behavior of integral invariant digital curvature estimators with the ball radius as a scale-space parameter. Figure 2, bottom row, shows the curvature values along the boundary of a "flower" shape, estimated with Eq. 5 (x-axis is curvilinear abscissa) with decreasing radii R (y-axis, from top to bottom). We observe that, around corner points, estimated curvatures tends to change according to the radius,

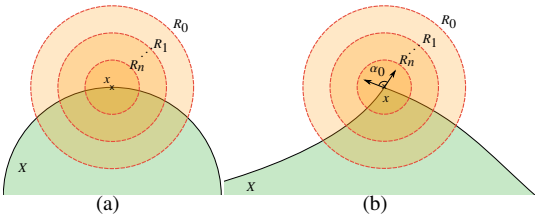


Figure 3: Scale-space analysis on a spherical shape (a) and a shape with a singularity (b).

while in smoother zones (low curvature), estimated curvatures are relatively insensitive to the radius.

To analyze this scale-space behavior, we propose to classify points of our input shapes into three categories: *edge*, *smooth* or *flat* regions.

3.1. Smooth and edge detection

Definition 2. *From Eq. 1, for any point x on the boundary of a Euclidean shape X in $\mathbb{R}^2$ (resp. $\mathbb{R}^3$), we define the* scale-space feature estimator $G_{X,x}(R)$ *(resp. $\mathcal{G}_{X,x}(R)$) as follows:*

$$G_{X,x}(R) \stackrel{def}{=} \frac{3\pi}{2R} - \frac{3A(R,x)}{R^3}, \quad \mathcal{G}_{X,x}(R) \stackrel{def}{=} \frac{8}{3R} - \frac{4V(R,x)}{\pi R^4}, \quad (13)$$

where R is the radius of a Euclidean ball centered on x, and $A(R,x)$ is the area (in 2D) and $V(R,x)$ is the volume (in 3D) of the intersection between this ball and the shape X.

In the following, we assume X to have a C^3 piecewise boundary, *i.e.* a smooth object with some singularities (locus of non-C^1 points). First, we study its behavior on a smooth part of the shape boundary (Fig.3-a). Then, we study its behavior at a singular point of the shape boundary (Fig.3-b).

Smooth case. In smooth case, *i.e.* when ∂X is C^3 at x, $G_{X,x}(R)$ is exactly the definition of curvature estimators of Eq. 2. We use another notation because at singular point x, $G_{X,x}(R)$ does not estimate the curvature but the angle between the two halves-tangents at x. Let us denote by κ_0 the curvature at x (Fig.3-a), and by H_0 its mean curvature in the 3D case. Since we are in the conditions of Eq.1, we know that $G_{X,x}(R) = \tilde{\kappa}(R,x) = \kappa_0 + O(R)$ and $\mathcal{G}_{X,x}(R) = \tilde{H}(R,x) = H_0 + O(R)$ when decreasing R to zero. So $G_{X,x}(R)$ gives us a constant term associated to the curvature at point x.

Singularity. We now consider a point x on a singularity (non$-C^1$ part) of ∂X (Fig.3-b). Following [15], we can perform Taylor expansion of the area and volume at x:

Proposition 1 (Eq. 12 from [15]). *Let X be a Euclidean shape in $\mathbb{R}^2$ (resp. $\mathbb{R}^3$) with at least C^3-smooth boundary piecewise, and let $x \in \partial X$ be a singularity. Then, if α_0 is the angle between halves-tangents at x and κ_- and κ_+ are left and right limit curvatures (resp. α_0 the opening angle of the sharp edge, H_- and H_+ the mean curvatures on either side of the edge), we have:*

$$A(R,x) = \frac{\alpha_0 R^2}{2} - \frac{\kappa_- + \kappa_+}{6} R^3 + O(R^4), \quad (14)$$

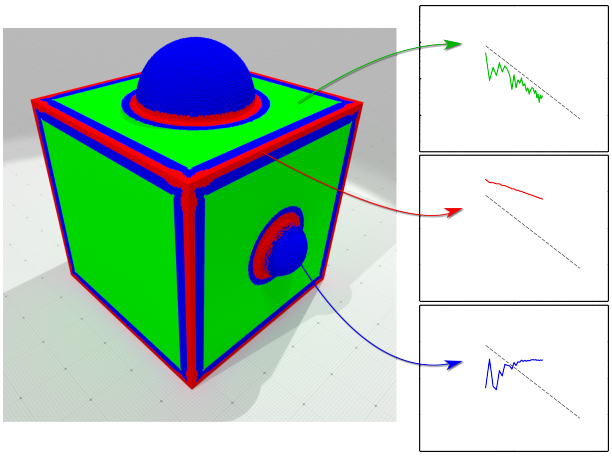


Figure 4: Left: Cube + Sphere of different size with feature mapped on surface elements. Green color is for flat parts, blue color for smooth parts and red color for edges. Right: Graphs in logarithmic scale of curvature value in a range of ball radii (from right to left, bigger to smaller radius) on a flat part (top), a smooth part (middle) and an edge (bottom).

$$V(R,x) = \frac{2\alpha_0 R^3}{3} - \frac{\pi(H_- + H_+)}{8} R^4 + O(R^5). \quad (15)$$

Therefore, quantities $G_{X,x}(R)$ and $\mathcal{G}_{X,x}(R)$ are functions of the ball radius R and the angle α_0 as:

$$G_{X,x}(R) = \frac{3}{2}\frac{1}{R}(\pi - \alpha_0) + \frac{\kappa_- + \kappa_+}{6} + O(R), \quad (16)$$

$$\mathcal{G}_{X,x}(R) = \frac{8}{3}\frac{1}{R}(1 - \frac{\alpha_0}{\pi}) + \frac{H_- + H_+}{2} + O(R). \quad (17)$$

In other words, both quantities $G_{X,x}(R)$ and $\mathcal{G}_{X,x}(R)$ are monomials of exponent -1 whose coefficient is related to α_0.

In conclusion, $G_{X,x}(R)$ (and $\mathcal{G}_{X,x}(R)$) has two different behaviors when we set the ball radius as a scale-space parameter whether we are on a smooth point or a singularity of the surface. At a point x on a smooth surface, the feature estimator $G_{X,x}(R)$ tends to the curvature at point x as R tends to zero. So the quantity is constant regardless of R (in fact, R has a limited impact on error terms). On a singularity, the feature estimator is a quantity that increases linearly as R decreases. In 3D, $\mathcal{G}_{X,x}(R)$ follows the same pattern.

So, for all surface elements, we compute the curvature on a range of decreasing ball radii (R_0 to R_n). To avoid issues in logarithmic scale with concave parts, we take the absolute value of curvature values. Figure 4 shows $\mathcal{G}_{X,x}(R)$ graphs for a given range of radii (abscissa), from right to left, on an edge (red graph) and a smooth region (blue graph). Flat region (green graph) will be discussed in the following section. The classification (color map) will be described in Section 3.3.

3.2. Influence of digitization

When digitizing $B_R(x) \cap X$, for a given ball radius R and a grid step h, there are infinitely many curves ∂X with different curvature values at x which lead to the same value $A(R,x)$ and thus the same quantity $G_{X,x}(R)$. It is also true in 3D. More precisely, we are interested in the possible curvature range when

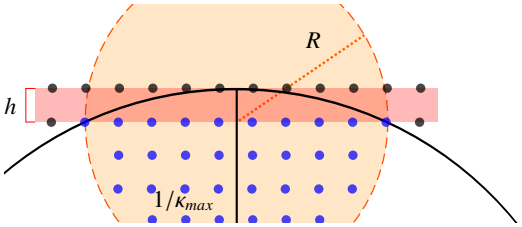


Figure 5: Digitization effect and notations for Property 1.

x lies in a flat or nearly flat area. The maximal curvature that may be misinterpreted as a flat region is given by the following Property:

Property 1. *Let X be a spherical Euclidean shape in $\mathbb{R}^2$ (resp. $\mathbb{R}^3$), R be the radius of ball and h be the grid step, for $x \in \partial X$, the maximal curvature value at x leading to the same quantity $G_{X,x}(R)$ (resp. $\mathcal{G}_{X,x}(R)$) is:*

$$\kappa_{max}(R,h) = \frac{2h}{R^2+h^2}, \quad \left(resp.\ H_{max}(R,h) = \frac{2h}{R^2+h^2}\right). \quad (18)$$

A geometrical illustration of the worst case inducing the bound is given in Figure 5.

As a consequence, if the quantity $G_{X,x}(R)$ is below $\kappa_{max}(R,h)$ (or $H_{max}(R,h)$ for the 3D case), we cannot decide if x belongs to a very soft smooth surface or a flat region. Hence, ball radius R (or the range of ball radius R in scale-space analysis) controls the *size* of the feature that can be recognized with respect to digitization artifacts. For instance in Figure 4, we observe the three principal cases in logscale: in green, all $\mathcal{G}_{X,x}(R)$ values are below the dashed line representing $H_{max}(R,h)$, which is characteristic of flat regions. In red, all values are above the dashed line with a slope -1, which is the expected behavior of edges. In blue, values are below the dashed line only for small radii R, which means that these radii are too small to capture the smoothness of the shape. For bigger radii, values are above the dashed line and constant, which means that this is a smooth region. As described in Section 3.3, the final classification will only consider radii for which values are above the $\kappa_{max}(R,h)$ curve (or $H_{max}(R,h)$ in 3D), points below are either considered as flat region or as outliers.

3.3. Distance to linear model based classification

We have two scale-space information we can use to define our classification: First, we have a flat region/outlier detector using $\kappa_{max}(R,h)$. Then, on remaining values, we know that $G_{X,x}(R)$ values and $\mathcal{G}_{X,x}(R)$ values behave differently on smooth region or at singular points as R decreases. Since 2D and 3D cases are similar, the following discussion applies indifferently for $G_{X,x}(R)$ values and $\mathcal{G}_{X,x}(R)$.

For a range of decreasing radii R_i, $0 \leq i \leq n$, we compute the feature estimator $G_{Z,p}(R)$ at point $p \in Bd(Z)$, the digital analog to $G_{X,x}(R)$. In a first pass, we remove outlier points from the graph of the function $G_{Z,p}(R)$ (i.e., points whose curvature value is below $\kappa_{max}(R,h)$). If there is not enough data (too many curvature values are below $\kappa_{max}(R,h)$), we classify the point as "flat" (green color in Fig.4). If we have enough data, we compute a least square fitting of the data, in logscale, with respect to a linear model of slope of 0 (so called "smooth model", the intercept being unknown) and a linear model of slope of -1 ("edge model", the intercept being unknown as well, see Eq. 16). For a given linear model of slope γ, the distance between the linear model e_γ and a range $\{G_{Z,p}(R_i), ..., G_{Z,p}(R_j)\}$ of n curvature values is given by

$$e_\gamma(G_{Z,p}(R_i), ..., G_{Z,p}(R_j)) = \min_{b\in\mathbb{R}} \left(\sum_{k=i}^{j} (Y_k - \gamma X_k + b)^2 \right), \quad (19)$$

with $X_k = \log R_k$ and $Y_k = \log(G_{Z,p}(R_k))$. Since we minimize a sum of quadratic terms, the value b^* for which Eq. 19 is minimal is simply:

$$b^* = \frac{\sum_{k=i}^{j} (\gamma X_k - Y_k)}{n}. \quad (20)$$

If the distance to the smooth model is lower than the distance to the edge model, we may classify the point as "smooth" (blue color in Fig.4), otherwise it may fall into the "edge" class (red color). For infinitely small radii and gridstep h, this classification perfectly captures the constant and hyperbolic behavior of curvature values, and it correctly decides whether the point of interest is singular or not.

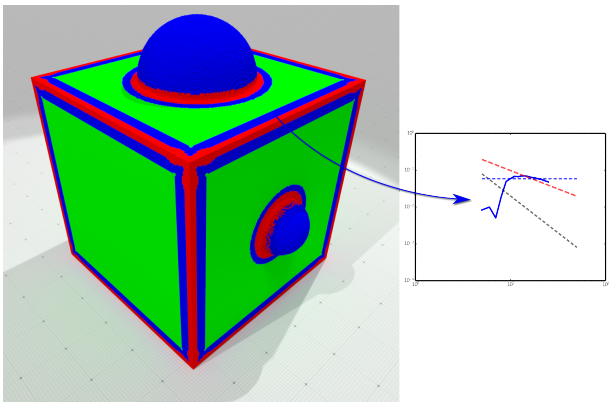


Figure 6: Graph of a model transition on a point near an edge.

3.4. Model transitions and overall classification

When dealing with noisy data, the ideal classification described above can be highly perturbed if noise induces high curvature values in the curvature profile for small radii. Beside such artifacts, for a finite range of radii, transitions may occur between classes. For instance, at a point which is close to an edge, this point could first be classified as "edge" for large radii, "smooth" for smaller ones and even "flat" if values fall above the $\kappa_{max}(R,h)$ value. This is illustrated in Figure 6 for a point close to an edge.

To recognize this behavior, we first introduce a new linear model of slope -2: the slope of $\kappa_{max}(R,h)$. If the distance to the

model of -2 slope (called *flat-model* hereafter) is lower than the distance to the smooth model and edge model, we can correct the classification to "flat". Then, we can evaluate the behavior of the three distances as the ball radius changes. Doing so, we can evaluate transitions in the linear fitting models. If a transition is detected (see Fig. 6), we decide to classify the point to the model which is minimal for the largest number of radii.

More formally, we define l_γ as the number of radii in the range $[R_0, R_n]$ for which the distance e_γ to the model of slope γ (for γ in $\{-2, -1, 0\}$) is minimal (compared to the other ones).

Finally, for a point p of the shape $Z \subset \mathbb{Z}^2$ or Z^3, we define our feature classifier as follows:

$$C_{Z,p}(R_0, R_n) = \begin{cases} \text{FLAT}, & \text{if } \forall 0 \leq i \leq n, G_{Z,p}(R_i) < \kappa_{max}(R_i, h) \\ & \text{or if } l_{-2} > \max(l_{-1}, l_0) \\ \text{SMOOTH}, & \text{if } l_0 > \max(l_{-1}, l_{-2}) \\ \text{EDGE}, & \text{otherwise.} \end{cases} \tag{21}$$

4. Experiments

We present an experimental evaluation of our feature estimator for 3D digital shapes. This Integral Invariant feature estimator (II) is implemented in the `DGtal` library [17]. We choose to compare our feature estimator with five other feature estimators from the literature we have adapted to digital data: three of them use a scale-space analysis [9, 6, 10] and two are "single-scale" [7, 8] but share common points with our method.

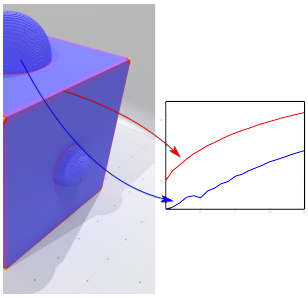
4.1. Comparative results

We first briefly describe the main characteristics of each feature estimator, then we discuss its pros and cons for digital shape analysis. Note that, apart from our method, all other estimators estimate features only from surfacic information (samples on ∂X). Although our method requires volumetric information (samples on X), our method keeps the same time complexity thanks to an optimization described in [1]. Input data are displayed on Figures 8, 9, 10 and 11, top row.

Figures 8 and 9 compare all feature estimators on noise free digitized shapes, while Figures 10 and 11 compare them on noisy versions of these digital shapes. For all methods, parameters—if any—were tuned to give best possible results for noise free data. These parameters are kept for noisy version of the shapes. For methods requiring a gridstep h (such as ours to define $\kappa_{max}(R, h)$), we simply consider $h = 1$ and encode this scaling factor in the object size for the experiment in the first row of Figure 7.

4.1.1. Clarenz's Feature Detector [7]

Clarenz *et al.* define a surface classification criterion based on the barycenter b and the covariance matrix (in their paper, they called them zero and first moments) of local surface patches $B_R(x) \cap \partial X$. They introduce a scale-space analysis of the length of the vector $\vec{xb}$ as a function of the ball radius. They show a quadratic scaling of this quantity on smooth regions of the surface, and a linear scaling close to edges. However, this scale-space property is not used in their classification methods, and it is also not evaluated experimentally. In our experiments, we observe these two behaviors, but they are hardly distinguishable (see graph of the right figure). Instead, they propose two methods that provide a smoothness score:



- the first one is based on the length of $\vec{xb}$ for a given ball radius. It provides a good smooth / non-smooth criterion on noise-free surfaces but not on noisy surface.
- the second one combines the previous feature score with the smallest eigenvalue (related to the normal) and the largest eigenvalue (related to the first principal curvature) of the covariance matrix of $B_R(x) \cap \partial X$ for a given R. This second estimator (used in the experiments) provides slightly better results on noisy shapes.

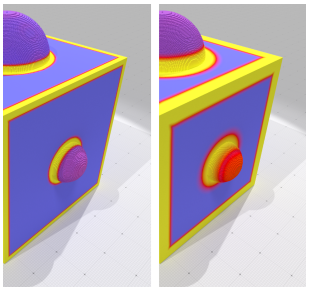
Note that both estimators require additional parameters (α and β) used to smooth the feature score.

In Figures 8 and 10, we show two radii R_1 and R_2 in the same range than others estimators. As expected, small radii detect small non-smooth region. Larger radii reinforce the non-smooth region detection but also consider small smooth region as non-smooth (see "union of spheres" object for example).

4.1.2. Mérigot's Voronoi Covariance Measure (VCM) [8]

Mérigot *et al.* propose a robust method to extract curvature information, sharp features and normal directions by convolving the covariance matrix of Voronoi cells on the object surface with a local kernel.

More precisely, to each point of the surface is associated a *Voronoi Covariance Measure* $\mathcal{V}_{X,R}(B(p, r))$, which depends on two parameters: the offset radius R dilates the input set (the distance function is more robust far from the surface), while the convolution radius r defines the Voronoi cells that are integrated to smooth the measure. Both parameters allow to limit the impact of noise in the input data while preserving geometrical information. To extract a feature score, authors compute a ratio of the eigenvalues of the convolved VCM at each point p: $r(p) \overset{def}{=} \frac{\lambda_2(p)}{\lambda_0(p)+\lambda_1(p)+\lambda_2(p)}$. The point is considered as a sharp edge when the ratio is greater than a threshold parameter T.

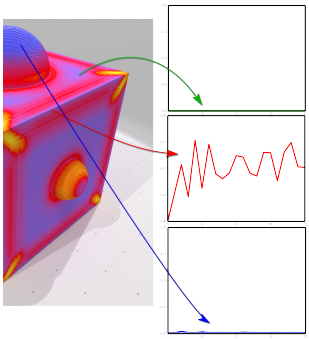


Experimentally, this method provides good results on both noise-free (see Fig. 8) and on noisy surfaces (Fig. 10). However it requires three parameters that are difficult to set for a large class of shapes. Note that the digital version of the VCM

has also some multigrid convergence properties for normal directions [18], and this is the implementation we used in our experiments.

4.1.3. Mellado's Growing Least Squares Analysis in Scale-Space [10]

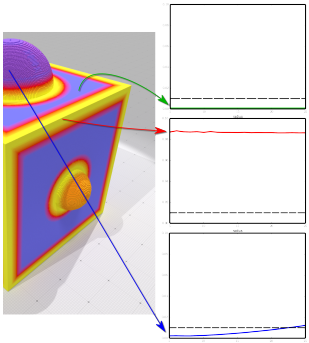
The method of Mellado *et al.* also provides a feature score rather than a feature classification as described above. More precisely, the shape is analyzed with a least squares spherical fitting approach. The scale-space parameter is the neighborhood size considered in the fitting. Then, following their notations, for each scale t, they fit an algebraic hyper-sphere and get the algebraic offset distance τ between p and the 0-isosurface, the unit normal η and the signed curvature κ of the hyper-sphere. Then, they compute a *geometrical variation* at a point p as defined by: $\nu(p,t) \stackrel{def}{=} \left(\frac{d\tau}{dt}\right)^2 + \left(t\frac{d\eta}{dt}\right)^2 + \left(t^2\frac{d\kappa}{dt}\right)^2$. Authors let the user choose what they consider a "feature" using this geometric variation. But they also provide a continuous feature function $f(p) \stackrel{def}{=} \int tanh(\nu(p,t))dt$ that differentiates regions with no geometrical variations (in blue color in Figs. 9 and 11) from those with high variations (in yellow).



Experimentally, this method seems to be less sensitive to noise but some artifacts appear on edges (see Fandisk object in Fig. 9).

4.1.4. Pauly [6]

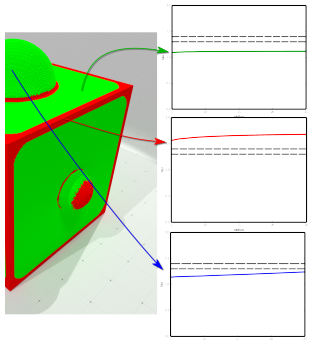
Like Clarenz *et al.* and Mérigot *et al.*, Pauly *et al.* use the eigenvalues of the covariance matrix at each point on the shape surface for a given neighborhood. They exploit all three eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_2$ by computing $\tau_i \stackrel{def}{=} \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}$ for a range of radii $\{R_i\}_i = 0..n$. Since these eigenvalues decrease as the curvature increases, τ_i will be higher on edges than on flat parts of the surface. To enhance this distinction, the weight $\omega(p)$ is defined as the number of times τ_i is greater than τ_{max} on the range of radii ($\omega(p) \stackrel{def}{=} Card\{\tau_i \geq \tau_{max} \,|\, 0 \leq i < n\}$). Such weight becomes large on feature parts (yellow color in Figs. 9 and 11) and low on non feature parts (blue color). Since this detector is related to the shape curvatures, it provides geometrically meaningful results. But the choice of τ_{max} is dependent on the shape geometry: it can consider small smooth parts as features (small sphere in Cube+Sphere object). Also, this estimator is highly sensitive to noise as we can see on Fig. 11.



4.1.5. Park's Tensor Voting Feature Extractor [9]

Park *et al.* compute the voting tensor at each point on the surface of the shape for a given neighborhood, then they compute eigenvalues of the resulting matrix. With a simple ratio on these eigenvalues, they get a feature weight quantity for each surface point at a given radius of neighborhood. They wish to classify the surface into two categories, *feature* (i.e. *edge* in fact) and *non-feature*.

The scale-space analysis comes with a threshold on those weights. They define two bounds: ω^- and ω^+; for all radii if the weight becomes greater than ω^+ the point is labeled as a "feature" (red color in Fig. 9), if the weight becomes lower than ω^- the point is labeled as a "non-feature" (green color), and if the variation of the weight is greater than τ times the last weight, they keep the last weight. The remaining points that are not classified will be clustered. Clusters containing less than ten points are classified as "non-feature", otherwise as "feature". From our own experiments, ω^- and ω^+ are really dependent on the shape geometry and the kind of "features" we want. At high curvature points, this analysis generally detects these points as features. However, when singularities have a smaller dihedral angle, bad classifications occur (on the Cube+Sphere object for example). On noisy data, with the same ω^-, ω^+ and τ parameters as for perfect data, it only detects high curvature regions as features.



4.1.6. Our feature estimator and discussion

All previous detectors discussed above fail to detect all feature regions at various scales. The difficulty lies in choosing parameters that detect small feature regions but wich are also robust enough on noisy shapes (see for example union-of-sphere objects). In our framework, we have ideal scale-space model of features. By analyzing the distance to each linear model as described in Section 3.3, our estimator can detect and properly classify edge regions from smooth and flat regions independently of their scale. Finally, no others parameters than a range of radii are required. It is important to note that the maximal radius R_0 of our range of ball radii is related to the lowest curvature of the smooth region we wish to detect. Otherwise, from Property 1, our estimator misclassifies it as a flat part.

Some strips around edge regions are wrongly classified as smooth (see Fig. 7-top for example): this artefact is due to the transition between a flat part and an edge part. If we analyze the transition, we see that the curvature follows a slope of -1 above $\kappa_{max}(R,h)$ (edge model) for large radii and ends with values below $\kappa_{max}(R,h)$ (flat model). Between these two states, the curvature must follow a slope of 0 (smooth model) to connect the two other states. This transition can be predominant close to a edge features. This artefact could be removed by measuring the geometric distance to a feature on the shape boundary,

and reclassify these false smooth zones into flat zones if this distance is small.

On noisy objects, our classification still detects edge, smooth and flat regions, but some edge artefacts may appear within smooth regions. This is due to the fact that the feature function is no more a straight line but more a polyline. These artefacts could be removed either by a better distance to models, or by pruning small regions (as in [9]). Remember that only raw results obtained by our method are presented. Therefore, it is clear that many post-processing could be added to improve results.

Finally, Figure 7-top displays the classification of our estimator on OctaFlower shape at different resolution (bounding boxes of digital objects are 256^3, 512^3 and 1024^3). Figure 7-bottom displays the classification of features on objects provided by scanners: a Stanford bunny, and digital snow microstructures acquired by X-ray microtomography. For these snow grains, the classification allows us to determine where the curvature estimation is not defined (on geometrical discontinuities).

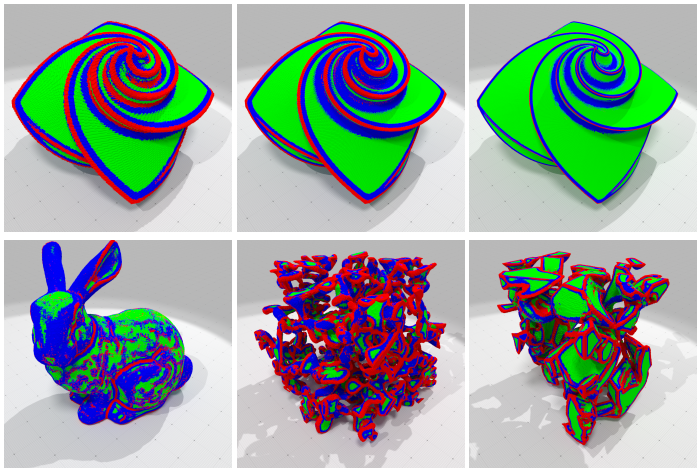


Figure 7: Results of our feature estimator on OctaFlower object at different resolutions (256^3, 512^3 and 1024^3), on a Stanford bunny and on snow microstructures acquired with X-Ray micro-tomography.

5. Conclusion

In this article, we have used integral invariant results from differential geometry to design a simple and a robust feature selection tool for digital objects. The proposed approach classifies digital surface elements into three categories: edge, smooth or flat. Since the proposed approach is based on local multiscale differential quantities, the final classification is locally adaptive and scale invariant in the sense that it can capture features at different scales on the same geometrical object.

For the specific case of digital surfaces, we have shown that our approach provides more accurate results compared to existing approach. Furthermore, the feature extraction is based on scale-space behavior of curvature estimators for which we have a multigrid convergence properties. As a consequence, quantities involved in the classification are still related to geometrical quantities defined on the underlying Euclidean object, and thus, a finer digitization implies a better classification.

References

[1] Coeurjolly D, Lachaud JO, Levallois J. Multigrid convergent principal curvature estimators in digital geometry. Computer Vision and Image Understanding 2014;129:27–41.

[2] Levallois J, Coeurjolly D, Lachaud JO. Parameter-free and multigrid convergent digital curvature estimators. In: Discrete Geometry for Computer Imagery (DGCI'2014); vol. 8668 of *LNCS*. Springer; 2014, p. 162–75.

[3] Guo Y, Bennamoun M, Sohel F, Lu M, Wan J. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 2014;36(11):2270–87. doi:10.1109.

[4] Lai YK, Zhou QY, Hu SM, Wallner J, Pottmann H. Robust feature classification and editing. IEEE Trans Visualization and Computer Graphics 2007;13:34–45.

[5] Yang YL, Lai YK, Hu SM, Pottmann H. Robust Principal Curvatures on Multiple Scales. In: Eurographics Symposium on Geometry Processing. 2006, p. 2–5.

[6] Pauly M, Keiser R, Gross M. Multi-scale Feature Extraction on Point-Sampled Surfaces. Computer Graphics Forum 2003;22(3):281–9.

[7] Clarenz U, Rumpf M, Telea A. Robust feature detection and local classification for surfaces based on moment analysis. IEEE transactions on visualization and computer graphics 2004;10(5):516–24.

[8] Merigot Q, Ovsjanikov M, Guibas L. Voronoi-based curvature and feature estimation from point clouds. Visualization and Computer Graphics, IEEE Transactions on 2011;17(6):743–56. doi:10.1109/TVCG.2010.261.

[9] Park MK, Lee SJ, Lee KH. Multi-scale tensor voting for feature extraction from unstructured point clouds. Graphical Models 2012;74(4):197–208.

[10] Mellado N, Guennebaud G, Barla P, Reuter P, Christophe S. Growing least squares for the analysis of manifolds in scale-space. Computer Graphics Forum 2012;31(5):1691–701.

[11] Gebal K, Bærentzen JA, Aanæs H, Larsen R. Shape analysis using the auto diffusion function. Computer Graphics Forum 2009;28(5):1405–13.

[12] Sun J, Ovsjanikov M, Guibas L. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. Computer Graphics Forum 2009;28(5):1383–92.

[13] Song R, Liu Y, Martin RR, Rosin PL. Mesh saliency via spectral processing. ACM Trans Graph 2014;33(1):1–17.

[14] Pottmann H, Wallner J, Yang YL, Lai YK, Hu SM. Principal curvatures from the integral invariant viewpoint. Computer Aided Geometric Design 2007;24(8-9):428–42.

[15] Pottmann H, Wallner J, Huang QX, Yang YL. Integral invariants for robust geometry processing. Computer Aided Geometric Design 2009;26(1):37–60.

[16] Digne J, Morel Jm, Souzani CM, Lartigue C. Scale space meshing of raw data point sets. Computer Graphics Forum 2011;30:1630–42.

[17] DGtal: Digital Geometry tools and algorithms library . `http://dgtal.org`. 2015.

[18] Cuel L, Lachaud JO, Thibert B. Voronoi-based geometry estimator for 3D digital surfaces. In: Proc. Discrete Geometry for Computer imagery (DGCI'2014). No. 8668 in LNCS; Springer; 2014, p. 134–49.

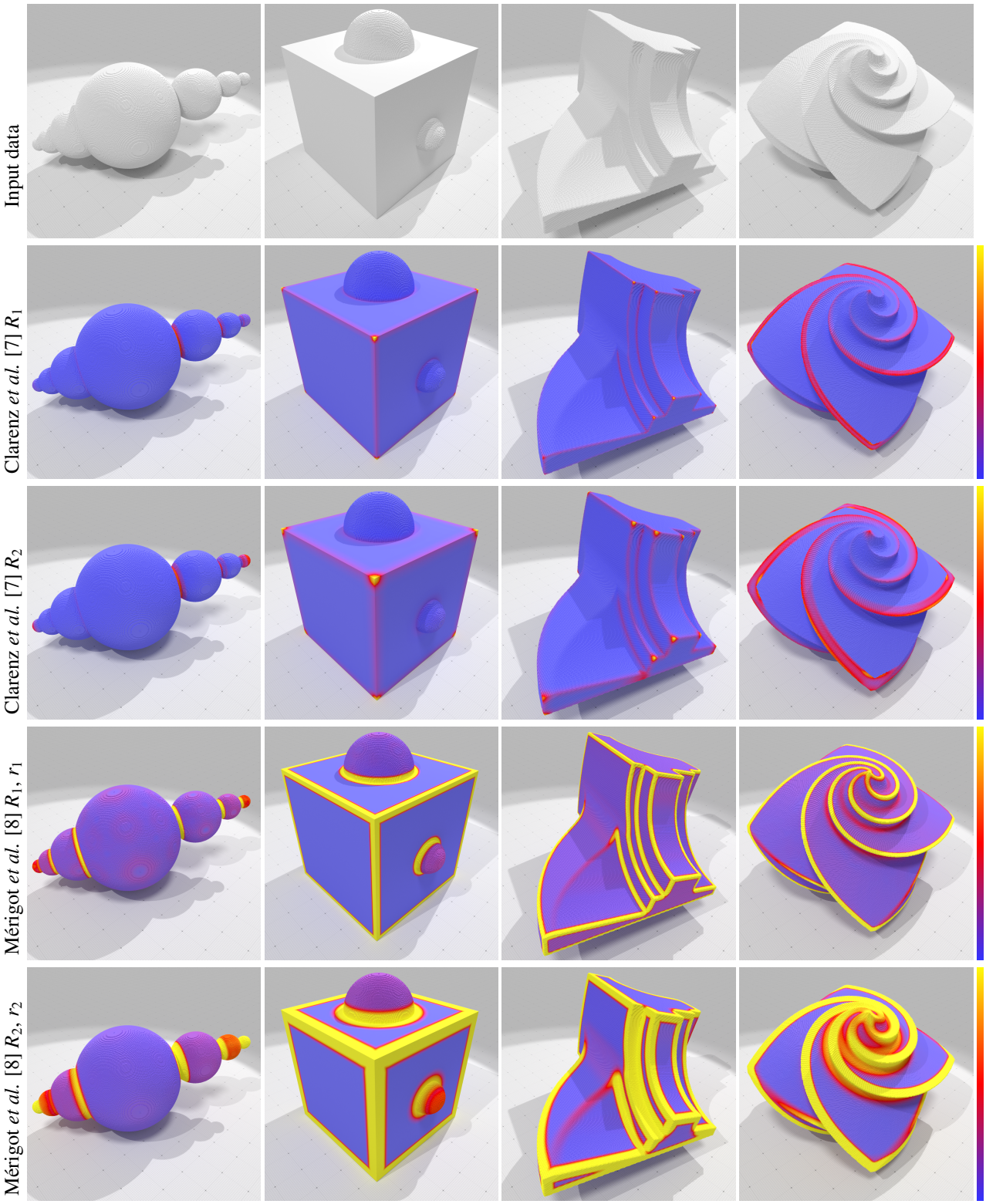


Figure 8: Evaluation of feature detectors on perfectly digitized shapes. SpheresUnion: $400 \times 200 \times 200$ voxels, CubeSphere: 200^3 voxels, Fandisk: 512^3 voxels, OctaFlower: 512^3 voxels. Parameters used for [7]: $R_1 = 10$, $R_2 = 22$, $\alpha = 1$, $\beta = 50$. Parameters used for [8]: $R_1 = 10$, $r_1 = 10$, $R_2 = 22$, $r_2 = 22$, $T = 0$.

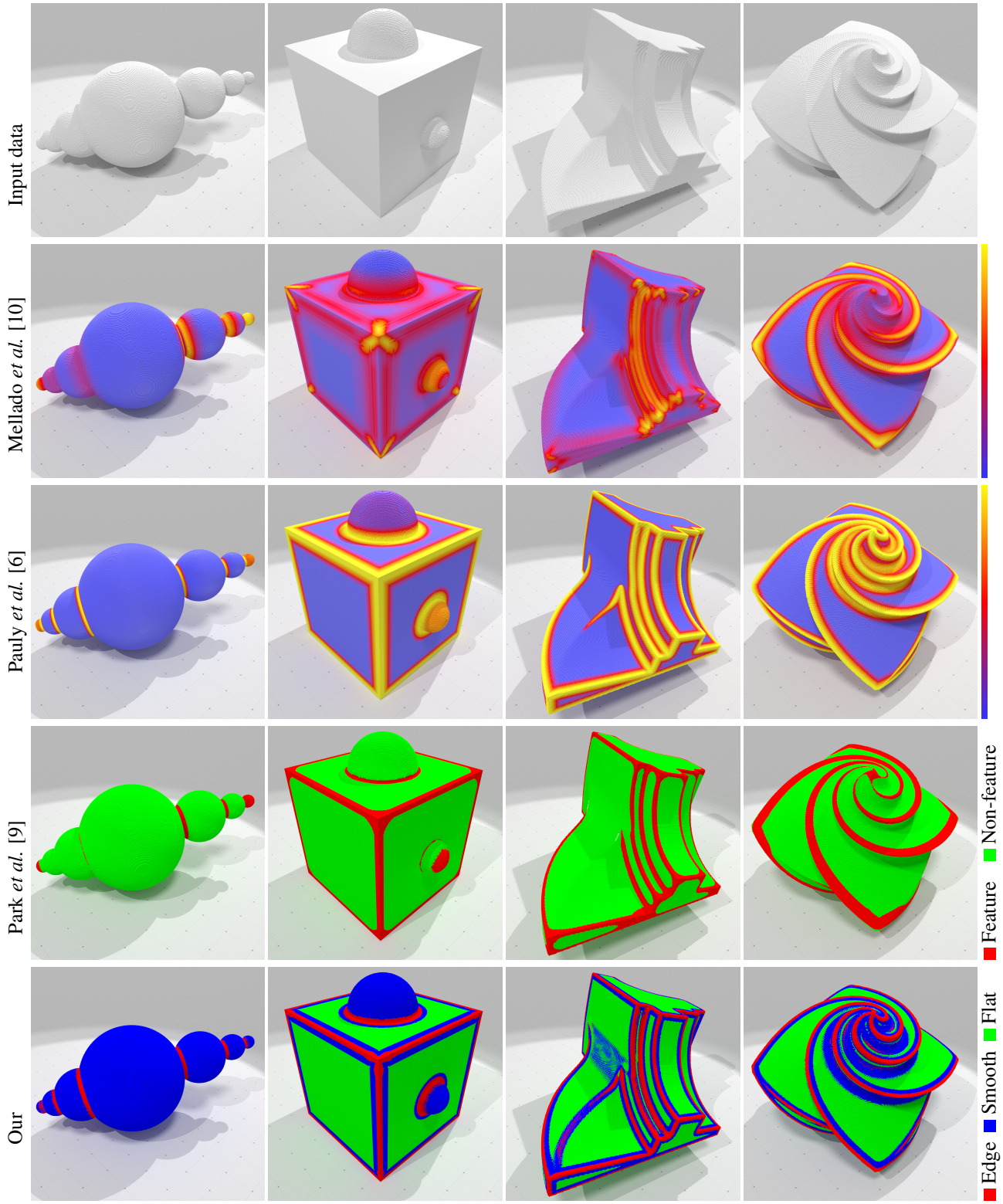


Figure 9: Evaluation of feature detectors on perfectly digitized shapes. SpheresUnion: $400 \times 200 \times 200$ voxels, CubeSphere: 200^3 voxels, Fandisk: 512^3 voxels, OctaFlower: 512^3 voxels. Parameters used for [10]: $r_{min} = 5$, $r_{max} = 25$. Parameters used for [6]: $r_{min} = 5$, $r_{max} = 25$, $\tau_{max} = 0.01$. Parameters used for [9]: $r_{min} = 5$, $r_{max} = 25$, $\omega_{min} = 1.4$, $\omega_{max} = 1.4$, $\tau = 1.2$. Parameters used for our algorithm: $r_{min} = 5$, $r_{max} = 25$.

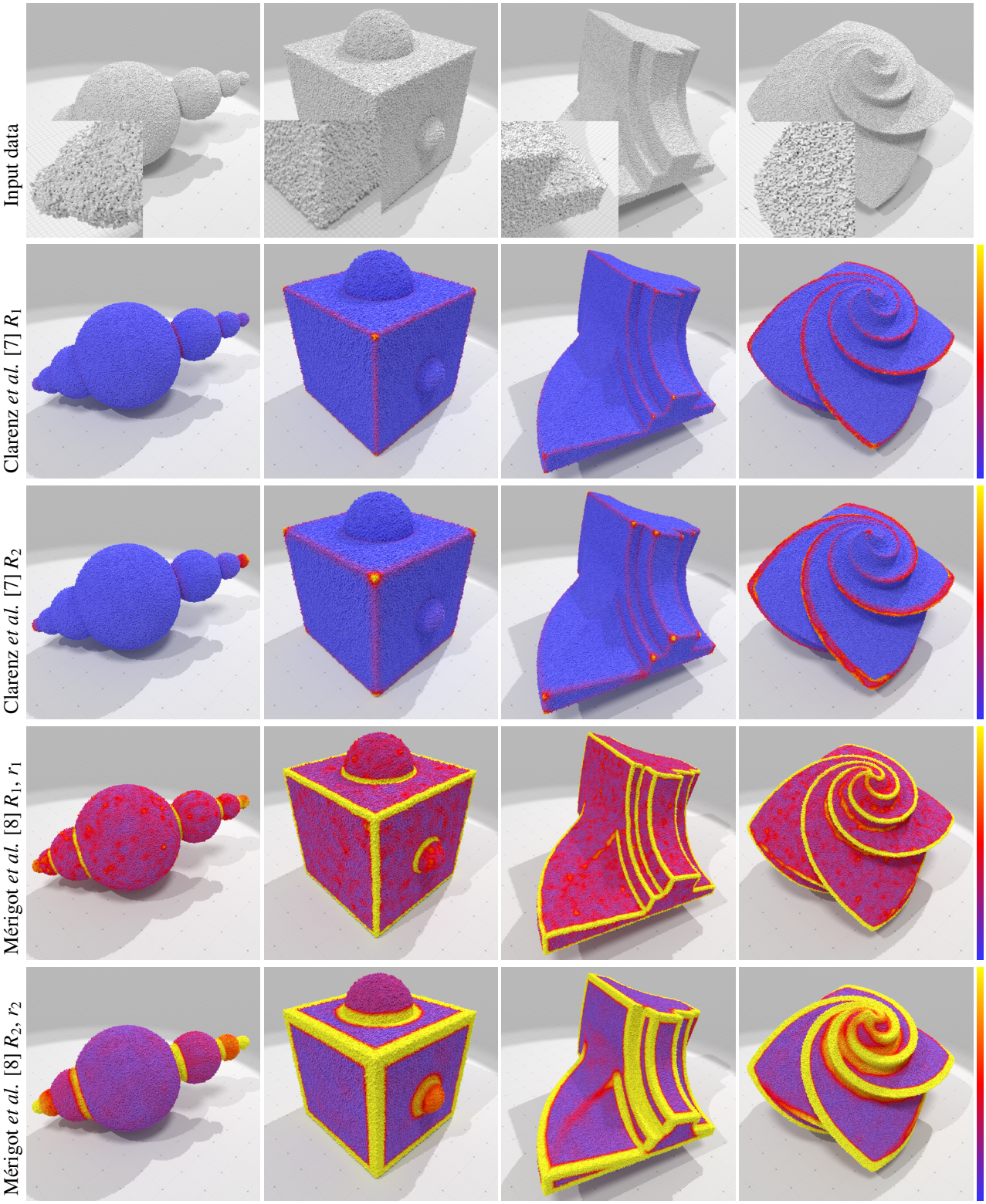


Figure 10: Evaluation of feature detectors on noisy approximations of digital shapes of Fig. 8. SpheresUnion: $400 \times 200 \times 200$ voxels, CubeSphere: 200^3 voxels, Fandisk: 512^3 voxels, OctaFlower: 512^3 voxels. Parameters used for [7]: $R_1 = 10$, $R_2 = 22$, $\alpha = 1$, $\beta = 50$. Parameters used for [8]: $R_1 = 10$, $r_1 = 10$, $R_2 = 22$, $r_2 = 22$, $T = 0.2$.

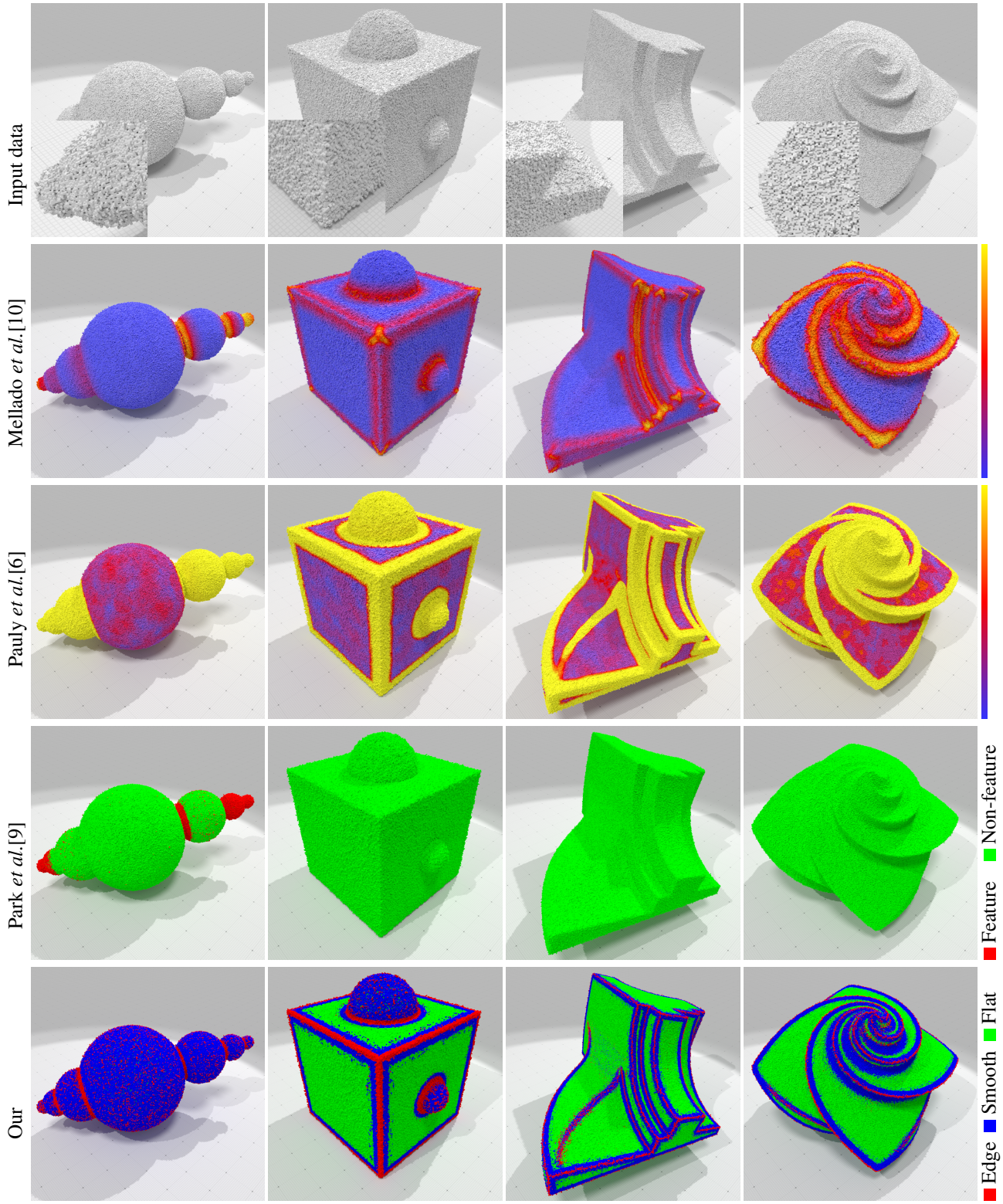


Figure 11: Evaluation of feature detectors on noisy approximations of digital shapes of Fig. 9. SpheresUnion: $400 \times 200 \times 200$ voxels, CubeSphere: 200^3 voxels, Fandisk: 512^3 voxels, OctaFlower: 512^3 voxels. Parameters used for [10]: $r_{min} = 5$, $r_{max} = 25$. Parameters used for [6]: $r_{min} = 5$, $r_{max} = 25$, $\tau_{max} = 0.01$. Parameters used for [9]: $r_{min} = 5$, $r_{max} = 25$, $\omega_{min} = 1.4$, $\omega_{max} = 1.4$, $\tau = 1.2$. Parameters used for our algorithm: $r_{min} = 5$, $r_{max} = 25$.